I. AMENDMENT

In the claims:

1. (previously presented) A communication circuit, comprising:

a network interface operable to receive a data packet having a header; and

a processor coupled to the network interface and operable

to perform a checksum operation on the received data packet, and if the checksum operation indicates that the data packet contains an error, to then change the received data packet to include an error indicator.

2. (original) The circuit of claim 1 wherein the data packet comprises a TCP data packet.

3. (original) The circuit of claim 1 wherein the data packet comprises an IP data packet.

4. (original) The circuit of claim 1 wherein the data packet comprises an UDP data packet.

5. (previously presented) A communication circuit, the circuit comprising:

a network interface operable to receive a data packet having a header;

a processor coupled to the network interface and operable to perform a checksum operation on the received data packet and operable to change the received data packet in response to the checksum operation;

wherein the header comprises a checksum field; and

wherein changing the received data packet comprises changing the checksum field.

6. (original) The circuit of claim 5 wherein:

the checksum field comprises a plurality of bits; and the processor is operable to change each bit in the checksum field to 0 if the received data packet passes the checksum operation.

7. (original) The circuit of claim 5 wherein:

the checksum field comprises a plurality of bits; and

the processor is operable to change each bit in the checksum field to 1 if the received data packet fails the checksum operation.

8. (original) A computer, comprising:

a network interface adapter having:

a media access controller having a network interface and operable to receive a data packet having a header; and

a processor coupled to the network interface and operable to perform a checksum operation on the received data packet and operable to change the received data packet in response to the checksum operation.

9. (original) The computer of claim 8, wherein the processor changes the received data packet in a first manner if the checksum operation is passed and changes the received data packet in a second manner if the checksum operation is failed.

10. (original) The computer of claim 9, further comprising a memory location operable to store the changed data packet.

11. (original) The computer of claim 10, further comprising a central processing unit coupled to the memory location and operable to process the changed data packet if is was changed in the first manner and operable to discard the changed data packet in the second manner.

12. (original) The computer of claim 11 wherein the central processing unit is further operable to perform a second checksum operation if the stored data packet was not changed in either the first or second manner.

13. (original) The computer of claim 10 wherein the memory location is a TCP/IP stack.

14. (previously presented) A method comprising:

generating a checksum from a data packet; and

adding an error indicator to the data packet if the checksum indicates that the data packet has an error.

15. (previously presented) A method comprising:

generating a checksum from a data packet;

changing the data packet in response to the checksum; and

receiving the data packet from a computer network via a network interface adapter having a media access controller.

16. (previously presented) A method comprising:

generating a checksum from a data packet;

changing the data packet in response to the checksum; and

wherein changing the data packet further comprises setting each bit in a checksum field in a header of the data packet to 0 if the data packet fails the checksum operation.

17. (previously presented) A method comprising:

generating a checksum from a data packet; and

changing the data packet in response to the checksum; and

wherein changing the data packet further comprises setting each bit in a checksum field in a header of the data packet to 1 if the data packet passes the checksum operation.

18. (previously presented) A method comprising:

generating a checksum from a data packet;

changing the data packet in response to the checksum; and

storing the changed data packet in a memory location.

19. (previously presented) The method of claim 18, further comprising:

analyzing the changed data packet;

processing the data packet if it was changed in a first manner; and

discarding the data packet if it was changed in a second manner.

20. (previously presented) The method of claim 18, further comprising performing a second checksum if the data packet was changed in manner other than a first or second manner.

21. (currently amended) A method, comprising:

generating a checksum from a data packet having a checksum field that includes a checksum value; and

modifying the checksum field in response to the checksum such that the checksum field no longer includes a checksum value.

22. (previously presented) A communication circuit, comprising:

a network interface operable to receive a data packet having a header; and

a processor coupled to the network interface and operable

to perform a checksum operation on the received data packet, and

if the checksum operation indicates that the data packet contains no error, to then modify the received data packet.

23. (previously presented) A method comprising:

generating a checksum from a data packet; and

modifying the data packet if the checksum indicates that the data packet has no errors.

24. (previously presented) The method of claim 23 wherein modifying the data packet comprises modifying a checksum field of the data packet.